

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA
BIOMEDICÍNSKÉHO
INŽENÝRSTVÍ**



**BAKALÁŘSKÁ
PRÁCE**

2017

**MATYÁŠ
DUDAŘ**



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

Katedra biomedicínské informatiky

Mikrokontrolér v úloze číslicové filtrace signálů

Microcontroller in the role of digital signal filtering

Bakalářská práce

Studijní program: Biomedicínská a klinická technika

Studijní obor: Biomedicínská informatika

Autor bakalářské práce: Matyáš Dudař

Vedoucí bakalářské práce: Ing. Jan Kauler, Ph.D.

Kladno 2017

PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci s názvem Mikrokontrolér v úloze číslicové filtrace signálů vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně 19.5.2017

.....

Matyáš Dudař

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce Ing. Janu Kaulerovi, Ph.D., za podporu, ochotu a trpělivost při tvorbě této bakalářské práce.

ABSTRAKT

Mikrokontrolér v úloze číslicové filtrace signálů

Prezentovaná práce se zabývá zpracováním a vykreslením elektrických biosignálů, specificky EMG – elektromyogramu. Pro práci se signálem je využito programového vybavení v prostředí Matlab, které vrací na základě frekvence vzorkování a požadované Logaritmicko – frekvenční charakteristiky koeficienty přenosu IIR.

Signál je v mikrokontroléru Arduino MEGA2560 převeden z analogového signálu do signálu digitálního a dále je filtrován pomocí diferenční rovnice získané z koeficientů přenosu. Výstupní signál je přes USB veden do PC a vizualizován v prostředí Matlab.

Klíčová slova

IIR filtr, EMG, Matlab, programovací jazyk C

ABSTRACT

Microcontroller in the role of digital signal filtering

Presented work is concerned with processing and depiction of electric biosignals, specifically EMG – electromyogram. Software of Matlab environment is used on work with the signal, which returns logarithmic-frequency coefficients of IIR transmission.

In the microcontroller, Arduino MEGA2560, is the signal converted from analog signal into digital and then is filtered with differential equation acquired from coefficients of IIR transmission. Output signal is transferred into computer and displayed in Matlab environment.

Keywords

IIR filter, EMG, Matlab, C programming language

Obsah

Seznam zkratek	5
1 Úvod	6
2 Zadání práce.....	7
2.1 Vymezení práce.....	7
2.2 Vytvoření hardwaru.....	7
2.3 Vypočítání koeficientů pro IIR filtr	7
2.4 Vytvoření programového vybavení.....	7
2.5 Záznam a vykreslení.....	7
3 Elektrické signály, jejich geneze, vlastnosti	8
3.1 Co je to EMG?.....	8
3.2 Geneze EMG signálů:	8
3.3 Vlastnosti EMG signálů	9
3.4 Zpracování a úprava biosignálů	10
3.4.1 Zesílení	10
3.4.2 Úprava signálu.....	11
3.4.3 Digitalizace.....	11
3.5 EKG.....	12
3.5.1 Popis EKG vlny	12
3.5.2 Umístění elektrod u EMG	13
3.6 EOG.....	14
3.6.1 Diagnostické využití EOG.....	14
4 Tvorba hardwaru pro zpracování signálu	15
4.1 Vlastnosti EMG signálu a z nich vyplývající požadavky na hardware.....	15
4.2 Vlastnosti mikrokontroléru Arduino MEGA2560 a z nich vyplývající požadavky na hardware	15
4.3 Návrh realizace hardwaru splňující požadavky bez podložení	16
4.3.1 Přístrojový zesilovač (Blok 1)	16
4.3.2 Pásmový selektivní zesilovač (Blok 2).....	17
4.3.3 Řešení hardwaru na nepájivém poli	17
5 Tvorba číslicových filtrů	18

5.1	Číslicové filtry obecně	18
5.2	Návrh číslicového filtru dolní propust	19
5.3	Návrh číslicového filtru typu horní propust	22
5.4	Návrh číslicového filtru typu pásmová propust	24
6	Tvorba programu na straně Arduina	27
6.1	Arduino obecně	27
6.2	Arduino IDE	27
6.3	Program na straně Arduina týkající se zpracování signálu	28
7	Program na straně Matlabu	30
7.1	Vykreslení výsledku zpracování signálu pro jeden kanál	31
8	Závěr	32
	Seznam použité literatury	33

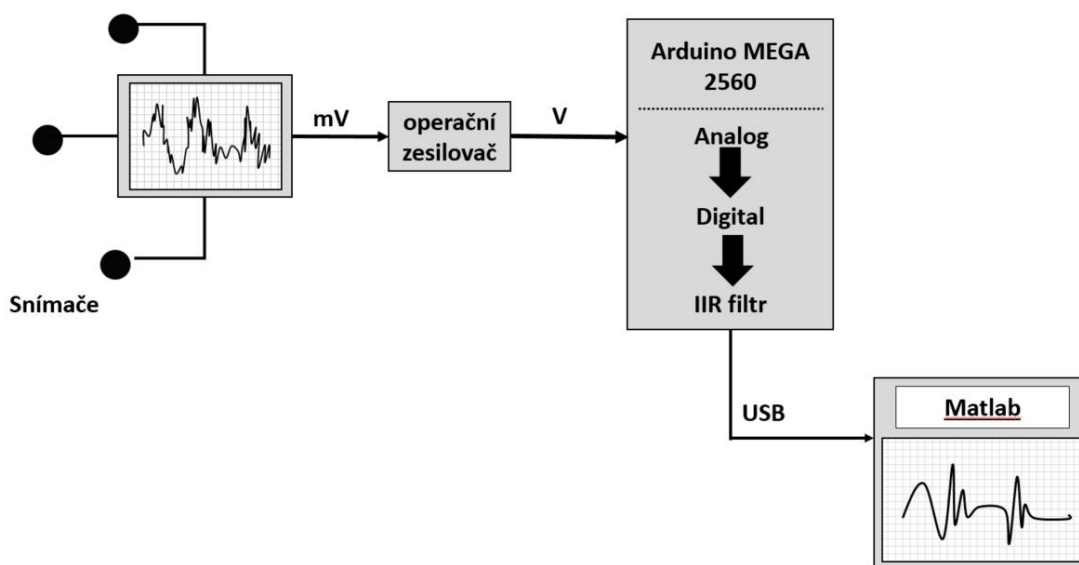
Seznam zkratek

Zkratka	Význam
EMG	Elektromyogram
EOG	Elektrookulogram
EKG	Elektrokardiogram
IIR	Infinite impulse response (filtr s nekonečnou impulzní odezvou)
HP	Horní propust
DP	Dolní propust
PP	Pásmová propust

1 Úvod

Na úvod této práce bych chtěl zmínit, že přístroje na měření biologických signálů jsou v současné době čím dál snáze dostupné. Nepoužívají se pouze v lékařství, ale i v běžném životě. Ovšem nevýhodou těchto uživatelsky dostupných řešení bývá neschopnost daného zařízení či programu v další práci se signálem. Přístroje pro práci s biologickými daty bývají stále cenově nedostupné a tvorba vlastního přístroje vyžaduje znalost analogových i digitálních zesilovačů, biologických signálů a programových prostředí.

Tato bakalářská práce se prvotně zabývá relativně jednoduchým zpracováním elektrického signálu, který je tvořen lidským tělem. Další částí této práce je vytvoření programových nástrojů ve vývojovém prostředí Matlab, které mají za úkol pouze po dosažení periody vzorkování a požadované frekvence řezu vypočítat koeficienty diferenciální rovnice. Tato diferenciální rovnice je potřeba k vytvoření digitálního filtru.



Obr 1.1 Triviální schéma vytvořeného zařízení pro zpracování elektrického signál

2 Zadání práce

2.1 Vymezení práce

Tato práce využívá základy v bakalářské práci T. Sládka, který již studium dokončil a ve své práci se také zaměřoval na využití EMG signálu. Jeho práce se však od této liší v dalším zpracování signálu. V práci T. Sládka se pracuje s nefiltrovaným signálem, který buď přesáhl určitou hranici a pak byla výstupem logická jednička, nebo naopak nepřesáhl danou hranici a výstupem byla logická nula. Tyto hodnoty pak využíval v ovládání robotické paže skrze mikrokontrolér ooPIC.

V této bakalářské práci se získaný EMG signál zpracovává na straně Arduina, což je mikrokontrolér, který slouží jako A/D převodník. Signál se dále přes diferenční rovnici, která se také nachází v kódu pro Arduino a zajišťuje filtraci signálu, odesílá přes USB do PC, kde je v prostředí Matlab zpracován (zaznamenán, vykreslen, odeslán dále etc.).

2.2 Vytvoření hardwaru

Vytvoření hardwaru pro snímání, zesílení a analogovou filtraci signálu. Tyto části jsou optimalizovány do signálu, který je dobře zpracovatelný pro mikrokontrolér Arduino MEGA2560.

2.3 Vypočítání koeficientů pro IIR filtr

Využití programového prostředí Matlab k určení koeficientů pro IIR filtr. Výstupem je diferenční rovnice, která je zakomponována v Arduinu.

2.4 Vytvoření programového vybavení

Vytvoření programu na straně mikrokontroléru Arduino MEGA2560 v jazyce C. Zde probíhá přijímání analogového signálu z hardwarové části a jeho převedení do signálu digitálního. Dále zde probíhá filtrace typu IIR pomocí dříve získané diferenční rovnice. Výstup je převeden do 8-bitových čísel pro zajištění efektivnějšího posílání dat přes USB.

2.5 Záznam a vykreslení

Vytvoření programu na straně Matlabu, který zajišťuje uložení hodnot do dále zpracovatelné podoby a následné vykreslení v grafu.

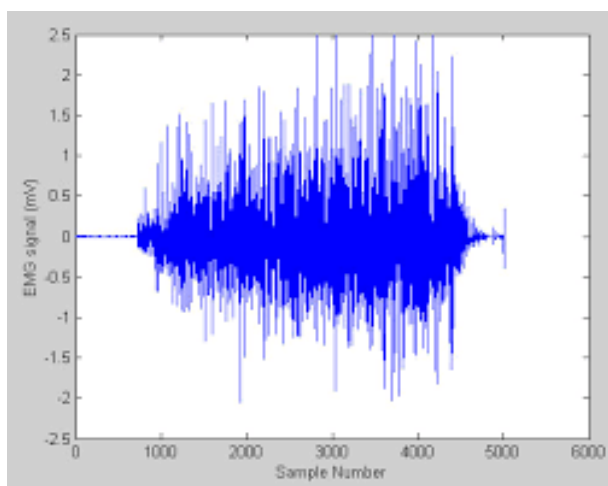
3 Elektrické signály, jejich geneze, vlastnosti

V této kapitole se z mnoha elektrických biosignálů změřím na popis EMG, se kterým jsem v průběhu práce počítal nejvíce. Na konci kapitoly jsou sekce věnované stručnému popisu ostatních elektrických biosignálů.

3.1 Co je to EMG?

Elektrický biosignál dostatečně silný, aby byl měřitelný na povrchu pokožky, produkovaný při svalové kontrakci (malou aktivitu vykonávají i svaly, které jsou v klidu) se nazývá elektromyografický signál neboli EMG signál a jeho záznam se nazývá elektromyogram. Jedná se stručně o rozdíl elektrických potenciálů.

Někdy se též zkratka EMG používá jako souhrnné označení oboru elektromyografie, které studuje správnou funkci svalů a neuronů, které je inervují, tím, že vyšetřuje přijímaný biosignál ze svalu.



Obr. 3.1 Ukázka EMG signálu

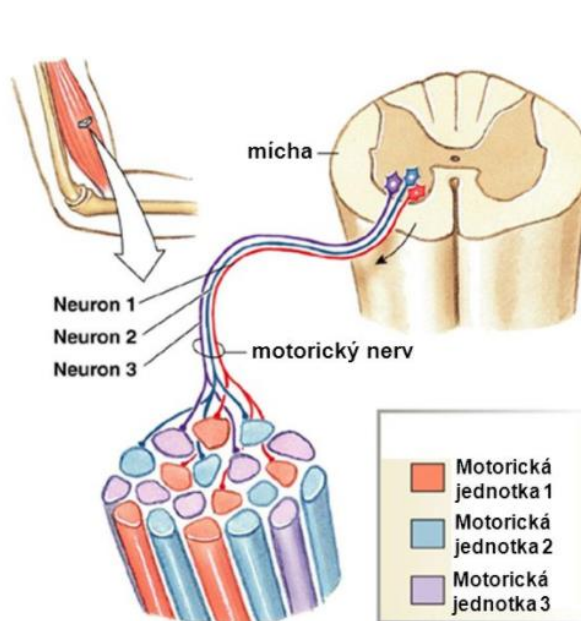
3.2 Geneze EMG signálů:

Pro vysvětlení geneze EMG signálu musíme začít vysvětlováním už od svalů jako takových.

Kosterní svaly jsou složeny ze skupin hustě pospojovaných nitkových buněk, dlouhých až 30 cm (dolní končetina – krejčovský sval), zvaných svalová vlákna. Svaly jsou ke kontrakci stimulovány nervovými impulsy přicházejícími z centrální nervové soustavy skrz míchu a motorické nervové kořeny, které obsahují jednotlivá motorická vlákna. Jedno motorické nervové vlákno může inervovat i několik svalových vláken. Spojení motorického nervového vlákna se všemi svalovými vlákny, které inervuje, bereme jako celek s označením motorická jednotka. Velikost motorické jednotky se

určuje podle množství svalových vláken, které inervuje jedno motorické nervové vlákno. Tato velikost se odvíjí podle umístění svalu na těle, respektive podle toho, jak jemná má být činnost tohoto svalu (např. obličejové svaly mají daleko menší motorické jednotky než svaly zádové). Proces, kdy se aktivují jednotlivé motorické jednotky, se nazývá nábor motorických jednotek. Síla kontrakce svalu je přímo úměrná množství zapojených motorických jednotek a dále frekvenci nervových impulsů (akčních potenciálů) přicházejících do každé motorické jednotky.

Akční potenciál vzniká, přestoupí-li depolarizační proud (vzruch) potenciál na úrovni prahu a vyvolá otevírání kanálů Na^+ , které vede ke zvýšení pozitivního náboje uvnitř buňky a k postupnému rozvoji akčního potenciálu. Vzrušení svalového vlákna podléhá ve většině případů jednoduchému zákonu "vše nebo nic". Nemusí tomu tak být vždy. Akční potenciál může vzniknout i součtem více jinak podprahových impulsů.



Obr. 3.2 Schéma vzniku EMG signálu

3.3 Vlastnosti EMG signálů

Amplituda elektrických impulsů přiváděných nervovými vlákny a dále vedených svalovými vlákny je velmi malá (v rozmezí 100microV až 90mV), ale protože jsou tyto impulsy vedeny souběžně mnoha vlákny, je vzniklý signál sumací všech akčních potenciálů, a proto je dostatečně velký na to, aby byl měřitelný na povrchu přiléhající pokožky pomocí páru elektrod. Zároveň je však utlumen okolními svalovými vlákny, která nejsou aktivní, a také vrstvou tuku a kůže.

Pokud tedy probíhá měření pomocí elektrod na povrchu kůže, amplituda EMG signálu se pohybuje v řádech mV. Tato problematika se dá obejít dvěma způsoby. Prvním

způsobem je jiný přístup ke snímání, a to pomocí vpichovaných elektrod přímo do svalové tkáně. Toto snímání přináší daleko lepší záznam, ale je to ergonomicky nepraktické řešení. Druhou možností je úprava signálu z povrchových elektrod. Přesněji řečeno zesílení signálu řádově 1000x, abychom s tímto signálem dokázali jednodušeji pracovat.

Frekvence EMG signálu není periodická podobně jako frekvence EKG. EMG signál vypadá spíše jako šum a rozsah jeho frekvencí se pohybuje od 20 Hz až do několika stovek Hz.

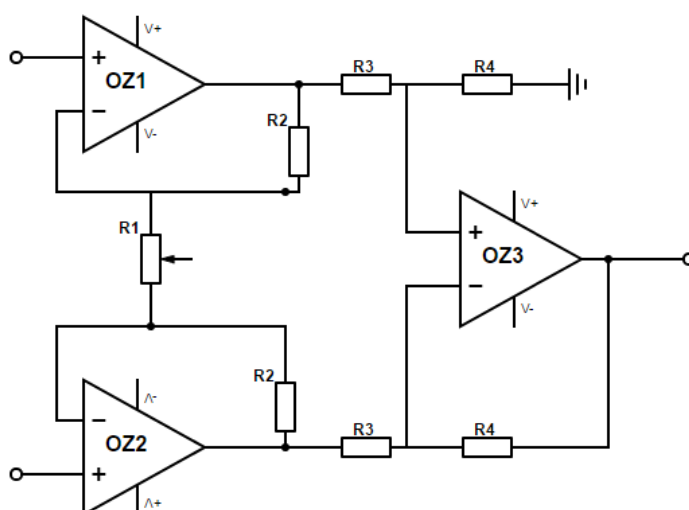
Z hlediska měření EMG signálu můžeme považovat tento signál za velice měkký zdroj napětí (velký vnitřní odpor), a proto není možné ho nějak zatížit (odebírat proud).

3.4 Zpracování a úprava biosignálů

3.4.1 Zesílení

Prvním krokem je zesílení přijímaného signálu na přijatelnou úroveň, se kterou se snáze pracuje. Pro signály EMG stačí nízkofrekvenční zesilovač (20 Hz - 20 kHz), který však musí mít vysoký vstupní odpor, aby se zdroj signálu nezatěžoval.

Dobrým řešením se jeví použití přístrojového zesilovače – v podstatě tří do sebe zapojených operačních zesilovačů. První dva zesilovače jsou zapojeny klasicky v neinvertující variantě (to zajišťuje vysoký vstupní odpor) a třetí zesilovač je zapojen jako diferenciální (rozdílový). Výhodou tohoto zapojení je malé vlastní rušení, velké rozdílové zesílení, potlačení souhlasného napětí a velký vstupní odpor.



Obr. 3.3 Obecné schéma přístrojového zesilovače

3.4.2 Úprava signálu

Přijímaný signál je potřeba ještě upravit, jinak bychom získali signál, který by při zobrazení byl špatně čitelný, jelikož by ukazoval i rušivé signály o jiných frekvencích. Například se jedná o rušení signálu ze sítě, jinak také označováno jako síťový brum, které má svou frekvenci 50 Hz.

Pro odstranění signálů s nežádoucí frekvencí použijeme zpravidla vhodný filtr. Pro síťové rušení se velmi hodí pásmová zádrž také známá jako notch filtr. Je to elektrický obvod, který má různá provedení. To budeme vybírat podle požadovaných parametrů. Požadavky námi zvoleného filtru jsou, aby filtr zastavil frekvence od 49 Hz do 51 Hz a všechny ostatní frekvence propustil. Toto bývá těžko realizovatelné, aniž by byly utlumeny frekvence kolem filtru.

V této situaci nám pomůže rozdílový zesilovač, který přijímá dva vstupy. Souhlasný signál těchto vstupů potlačí a zesílí pouze signály rozdílné. V našem případě je u obou vstupů síťový brum, který bude tímto zesilovačem potlačen.

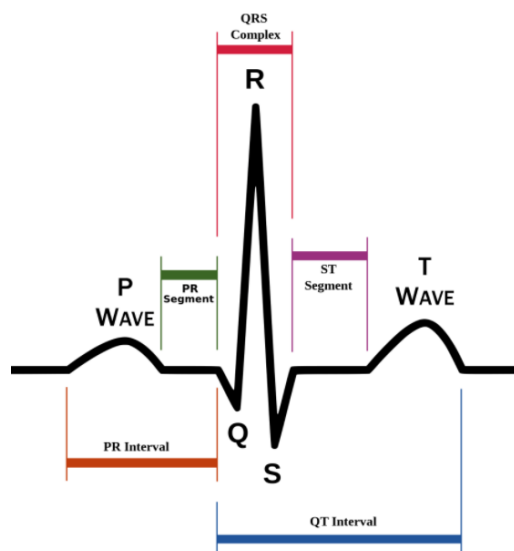
3.4.3 Digitalizace

Po zesílení a filtraci signálu od síťového rušení je nutné převedení analogového signálu do signálu digitálního. K tomu využijeme mikrokontroléru Arduino MEGA2560, který má už v sobě A/D převodník zakomponován. Signál je tedy po odstranění síťového rušení posílán do analogového vstupu mikrokontroléru.

3.5 EKG

Další elektrický biosignál, který se dá snadno zpracovávat tímto zapojením je EKG. Stačilo by zvolit jinou frekvenci řezu pro filtrování signálu. Ale i bez této změny je EKG vidět, i když s utlumenými hodnotami.

EKG je zkratka pro elektrokardiogram. Jedná se tedy o snímání signálu aktivity srdečního svalu.



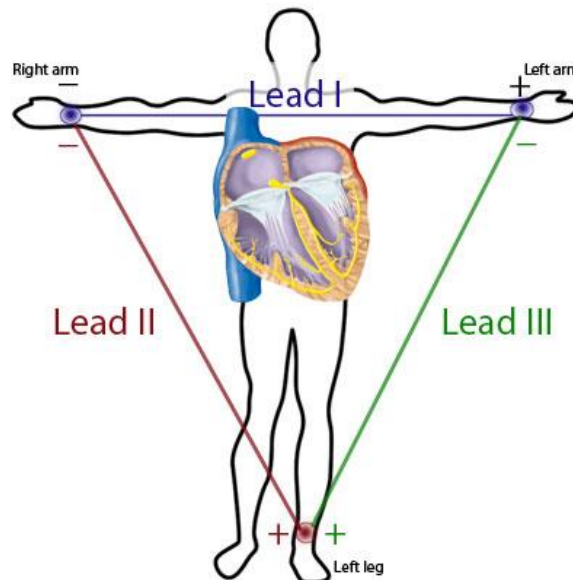
Obr 3.4 Obecný popis jednoho cyklu EKG

3.5.1 Popis EKG vlny

Vlna P vzniká při depolarizaci síní. Interval P-Q reprezentuje čas potřebný pro depolarizaci síní a dobu šíření vzruchu atrioventrikulárním vedením. Vlna Q vyjadřuje depolarizaci komorové přepážky. Vlna R označuje depolarizaci přední, zadní a bočních stěn srdečních komor – tedy začátek systol komor. Interval P-R představuje dobu vedení Hisovým svazkem. Interval S-T odpovídá období ukončování depolarizace komor. Vlna T reprezentuje rychlou repolarizace komor. Q-T interval odpovídá elektrické systole komor.

3.5.2 Umístění elektrod u EMG

Pro snímání EKG je důležité jinak zapojit elektrody na těle. U EMG stačilo mít elektrody na jedné ruce, ale u EKG musíme elektrody zapojit na obě zápěstí a uzemňující elektrodu k levému kotníku. Důvodem je to, že je důležité, aby napětí z elektrod procházelo přes srdeční oblast. Zapojení obou zápěstí a kotníku se také říká Einthovenův trojúhelník.

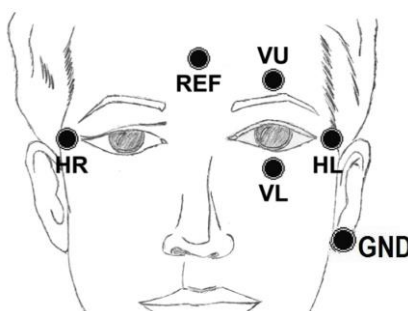


Obr 3.5 Einthovenův trojúhelník

3.6 EOG

Elektrookulografie slouží k měření polohy oka pomocí elektrických potenciálů snímaných povrchovými elektrodami. Pro měření horizontální složky jsou elektrody po stranách oka a pro vertikální složky jsou elektrody umístěny pod a nad okem.

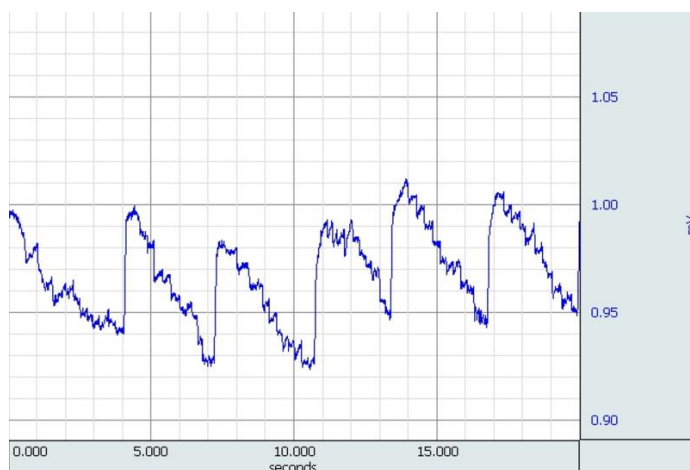
Potenciál dipólu rohovka – sítnice závisí na poloze oční osy. Pokud je tedy pohled upřen vpřed, zmíněný dipól je umístěn symetricky mezi elektrodami a výsledný signál EOG je nulový. Při pohledu vpravo se stane rohovka pozitivní v blízkosti pravé elektrody, která se také stává pozitivní.



Obr. 3.6 Umístění elektrod pro měření EOG

3.6.1 Diagnostické využití EOG

EOG se dá využít pro mnoho vyšetření. Například pro vyhodnocení funkčnosti okohybných svalů, asymetrie oka nebo vysunutí oka z očníce. Patologický EOG je také u některých cévních poruch a retinopatií (cukrovka, vysoký krevní tlak, záněty cév, ...). Zvláštní EOG je možné zaznamenat při čtení. Oči se pohybují podél řádek ve formě rychlých skoků, které trvají od 0,15 s až 0,5 s. Na konci řádku se uskutečňuje jeden rychlý skok na začátek následující řádky. Amplituda a frekvence závisí na formátu a obtížnosti textu.



Obr. 3.7 EOG naměřené při čtení

4 Tvorba hardwaru pro zpracování signálu

4.1 Vlastnosti EMG signálu a z nich vyplývající požadavky na hardware

Pro připomenutí bodu 3.2, hlavním požadavkem na hardware je, aby zesilovač měl velký vstupní odpor, a tak nedošlo k téměř žádnému zatížení zdroje EMG signálu. V případě nedostatečně velkého vstupního odporu by napětí kleslo k nule.

Dalším požadavkem je zesílení. Jelikož je napětí elektrických biosignálů včetně EMG v řádech milivoltů a my potřebujeme signál v řádech voltů, musíme zesílit přijímaný signál přibližně 1000krát.

Dalším krokem je potřeba odstranit síťové rušení s frekvencí 50 Hz, které je naindukované na svodech.

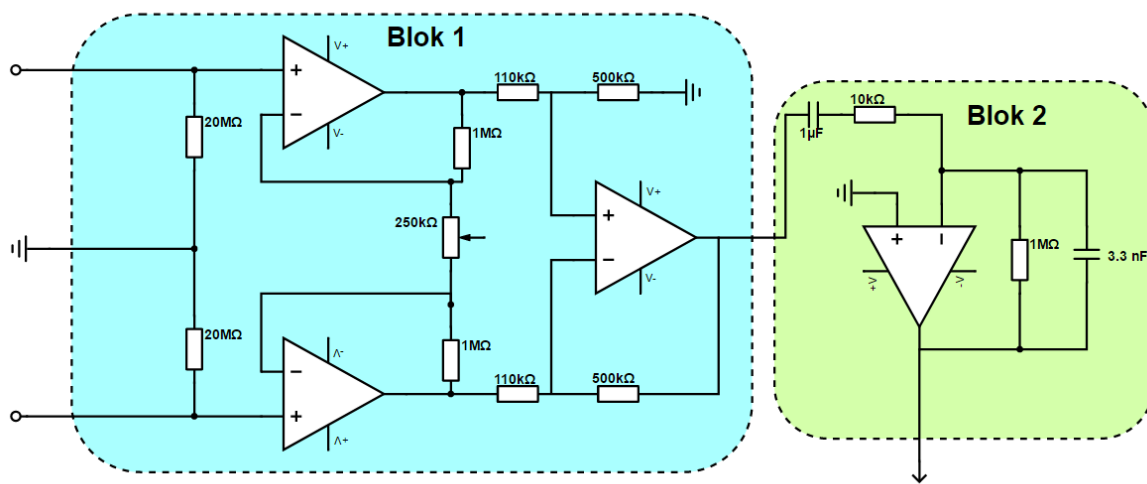
4.2 Vlastnosti mikrokontroléru Arduino MEGA2560 a z nich vyplývající požadavky na hardware

Zprvė musím zmínit velikost napětí, kterou Arduino přijme. Ta je jak na vstupech i výstupech v rozmezí 0-5 V. Pro EMG, se tedy bude muset signál podložit, aby nešel do záporných hodnot.

Co se týká rychlosti vzorkování, zde žádné problémy nejsou, jelikož Arduino zvládá i daleko větší frekvence, než je požadovaných 200 Hz.

4.3 Návrh realizace hardwaru splňující požadavky bez podložení

Níže je vidět schéma hardwaru pro zpracování elektrických biosignálů. Schéma je pro přehlednost rozděleno do dvou bloků. Během tvorby této BP jsem pracoval pouze se signálem, který byl složen pouze z jeho kladné části, abych nemusel zapojovat podložení. Ořezání signálu je provedeno pomocí diody, která je zapojena mezi přístrojovým selektivním zesilovačem (Blok 2) a vstupem do mikrokontroléru Arduino MEGA2560.



Obr.4.1 Schéma hlavní části hardwaru

4.3.1 Přístrojový zesilovač (Blok 1)

Jde o normální přístrojový zesilovač určený zesílení rozdílového napětí a potlačení napětí souhlasného. Operační zesilovače 1 a 2 slouží jako invertující napěťové zesilovače a 3. operační zesilovač je rozdílový (diferenciální). Na svody 1 a 2 jsou připojeny elektrody nalepených na koncích svalu.

Toto zapojení je fungující, jelikož zajišťuje dostatečně velké zesílení, velký vstupní odpor a zároveň velké potlačení souhlasného signálu. Potlačení síťového rušení je podpořeno propojením referenční elektrody, která je připojena v přibližném geometrickém středu subjektu generujícího EMG, se zemí.

Protože velikost amplitudy EMG je u každého trochu jiná, je možnost ovlivnit zesílení pomocí potenciometru, který je ještě před rozdílovým zesilovačem.

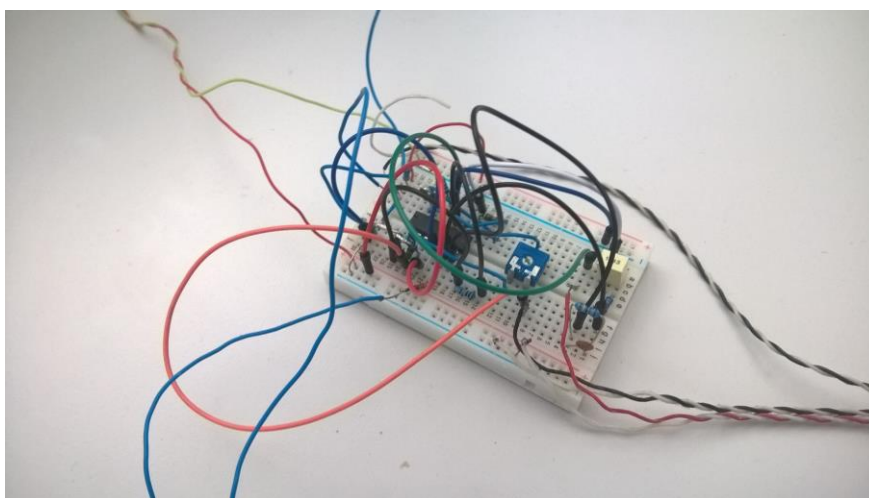
Použitý operační zesilovač je integrovaný obvod TL074CN, který obsahuje v pouzdru čtyři operační zesilovače.

4.3.2 Pásmový selektivní zesilovač (Blok 2)

Toto zapojení má za úkol zesílit pouze určitý rozsah frekvencí signálu, zatímco zbylé frekvence signálu potlačit. V tomto případě jsou hodnoty použitých součástek voleny tak, aby hraniční hodnoty propouštěných frekvencí byly přibližně 20 Hz a 200 Hz.

4.3.3 Řešení hardwaru na nepájivém poli

Před tištěným spojem bylo hardwarové řešení vytvořeno na nepájivém poli z důvodu jednoduché úpravy případných chyb. Tři výstupní vodiče jsou pro elektrody, další tři pro napájení a zem a poslední dva jako výstup do Arduina a zem Arduina.



Obr.4.2 Hardwarové řešení na nepájivém poli pro jeden kanál.

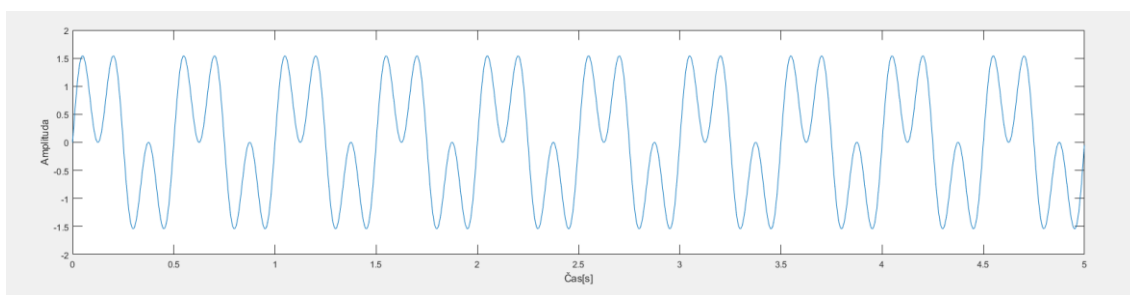
Schémata tištěného spoje jsou k nalezení na CD příloze.

5 Tvorba číslicových filtrů

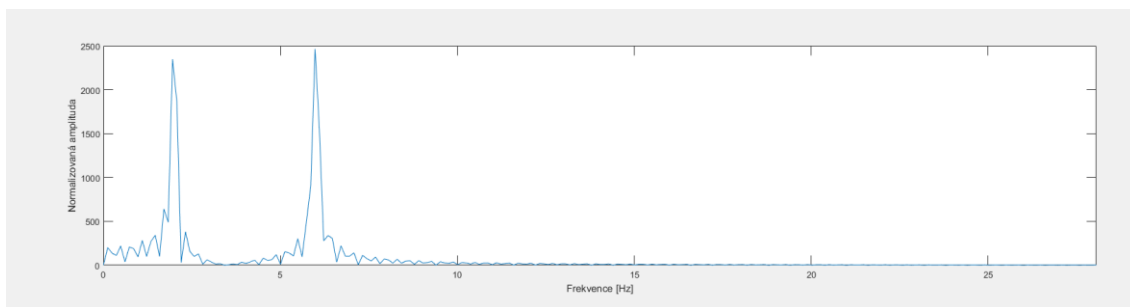
I když pro EMG signál tvoříme číslicový filtr typu dolní propust (DP), ze zadání práce plyne i vytvoření filtrů typu horní propust (HP) a pásmová propust (PP). První část této kapitoly věnuji obecným informacím ohledně filtrů.

5.1 Číslicové filtry obecně

Filtrace signálu patří mezi základní metody zpracování signálu. Jedná se o odebrání (přesněji potlačení) určitého rozsahu frekvencí ze signálu, který chceme filtrovat. Pro jednoduchou představu si vytvoříme signál, který se bude skládat ze dvou frekvencí $f_1 = 2$ Hz a $f_2 = 6$ Hz.



Obr.5.1 Časová doména příkladového signálu

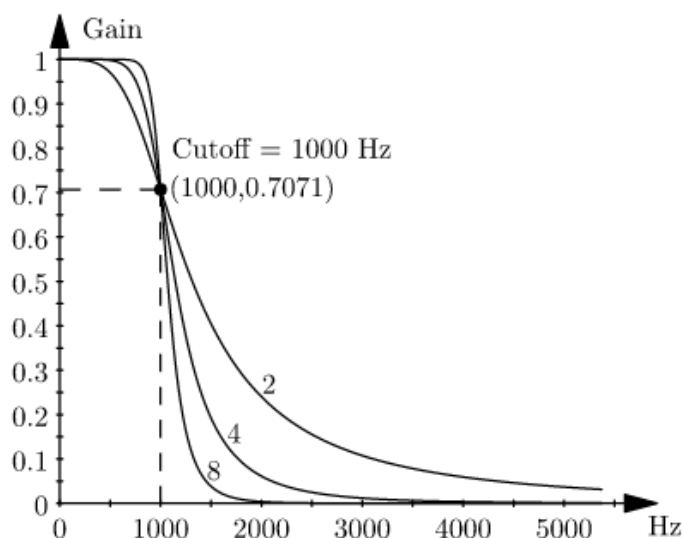


Obr.5.2 Frekvenční doména příkladového signálu

Ve frekvenční doméně můžeme vidět dvě hodnoty, které podle osy X odpovídají zadaným frekvencím příkladového signálu. Toto spektrum jsem vytvořil pomocí FFT (Fast Fourier Transform). Kdybychom chtěli z tohoto signálu odfiltrovat vyšší frekvence, které by mohly odpovídat i šumu, použili bychom filtr typu DP. Ten potlačuje frekvence vyšší, než je f_{cut} (cutoff frekvence – hodnota frekvence, se kterou se počítá při tvorbě požadovaného filtru).

5.2 Návrh číslicového filtru dolní propust

Dolní propust propouští, jak už je z názvu domyslitelné, pouze frekvence s nižší hodnotou, než je f_{cut} .



Obr.5.3 Příklad Butterworth filtru typu DP

Návrh číslicového filtru jsem provedl v programovém prostředí Matlab, ve kterém jsem si vytvořil vzorce odpovídající literatuře. Zaměřil jsem se na Butterworth filtr druhého řádu. Následující rovnice je stejná u všech typů Butterworthových filtrů druhého řádu.

$$H(p) = \frac{1}{p^2 + \sqrt{2}p + 1}$$

Pro požadovanou DP zvolíme převodní vztah

$$p \Rightarrow \frac{p}{\omega_p}$$

Po převodu a úpravě získáváme

$$\frac{\omega_p^2}{p^2 + \sqrt{2}p\omega_p + \omega_p^2}$$

Nyní využijeme bilineární transformaci a opět nahradíme p

$$p = \frac{2}{T_{vz}} \frac{z-1}{z+1}$$

Obecný vzorec je tedy

$$\frac{\omega_p^2}{\left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right)^2 + \sqrt{2} \left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right) \omega_p + w_p^2}$$

Do tohoto vzorce se místo ω_p dosadí hodnota spočtené analogové osy kmitočtu podle zvolené f_{cut} .

$$\omega = 2\pi f_{cut}$$

$$\omega_p = \frac{2}{T_{vz}} \operatorname{tg}\left(\frac{\omega T_{vz}}{2}\right)$$

Tyto vzorce jsem použil pro vytvoření diferenciální rovnice. Výsledné koeficienty jsou součástí digitálního filtru na straně mikrokontroléru Arduino MEGA2560.

Návrh kódu pro DP v programovém prostředí Matlab s následným výstupem

```
clearvars
clc

z=tf('z');

% Perioda vzorkovani
Tvz = 0.005;

% Frekvence rezu [Hz]
fcut = 4;

% Vypocet omegy s digitalni osou
omega = fcut*2*pi;

% Vypocet omegy s analogovou osou
omega_a = (2/Tvz)*tan(omega*Tvz/2);

% Bilinearni transformace
p = ((2/Tvz)*((z-1)/(z+1)));

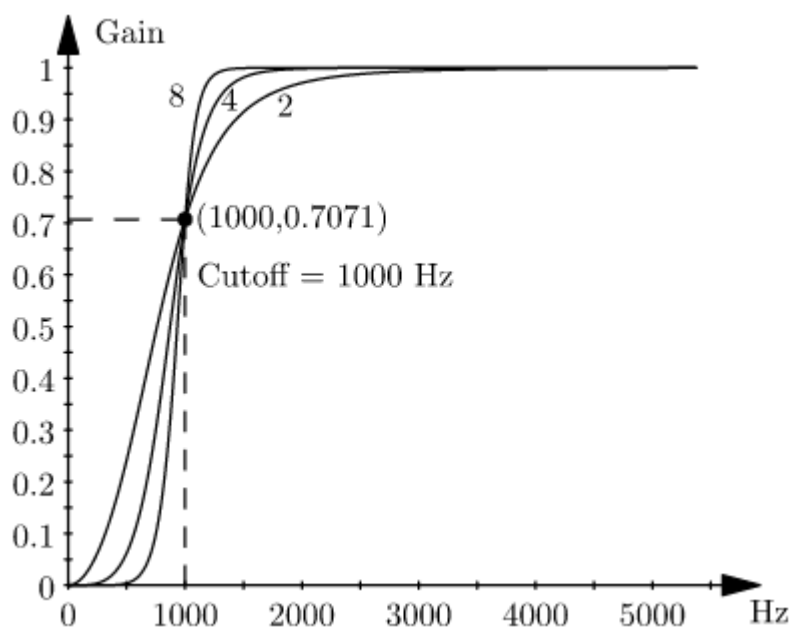
% Filtr .... H(p) = 1 / (p^2 + sqrt(2)*p + 1)
% Dolni Propust .... p => p / omega_a
Hz = (omega_a)^2/((p^2)+sqrt(2)*p*omega_a+(omega_a^2))
```

Hz =

$$\frac{633.3 \, z^3 + 1900 \, z^2 + 1900 \, z + 633.3}{1.749e05 \, z^3 - 1.439e05 \, z^2 - 1.723e05 \, z + 1.464e05}$$

5.3 Návrh číslicového filtru typu horní propust

Filtru typu horní propust je filtr, který propouští frekvence s hodnotou vyšší, než je u f_{cut} .



Obr.5.4 Příklad Butterworth filtru typu HP

Při návrhu číslicového filtru typu horní propust opět zvolíme Butterworthův filtr druhého řádu. Tedy

$$H(p) = \frac{1}{p^2 + \sqrt{2}p + 1}$$

Převodní vztah pro HP

$$p \Rightarrow \frac{\omega_p}{p}$$

Jak si můžeme všimnout, převodní vztah pro HP je převrácenou hodnotou převodního vztahu pro DP. Obecný vzorec se tedy také nebude tolik lišit.

$$\frac{\left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right)^2}{\left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right)^2 + \sqrt{2} \left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right) \omega_p + \omega_p^2}$$

Návrh kódu pro HP v programovém prostředí Matlab s následným výstupem

```
clearvars
clc

z=tf('z');

% Perioda vzorkovani
Tvz = 0.005;

% Frekvence rezu
fcut = 4;

% Vypocet omegy s digitalni osou
omega = fcut*2*pi;

% Vypocet omegy s analogovou osou
omega_a = (2/Tvz)*tan(omega*Tvz/2);

% Bilinearni transformace
p = ((2/Tvz)*((z-1)/(z+1)));

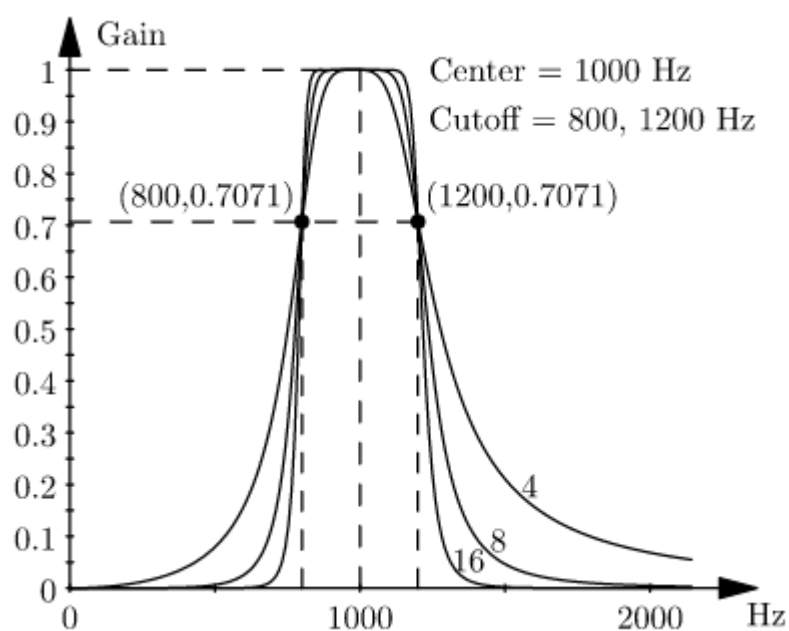
% Filtr .... H(p) = 1 / (p^2 + sqrt(2)*p + 1)
% Horni Propust .... p => omega_a / p
Hz = (p^2)/((p^2)+sqrt(2)*p*omega_a+(omega_a^2))
```

Hz =

$$\frac{160000 z^5 + 160000 z^4 - 320000 z^3 - 320000 z^2 + 160000 z + 160000}{1.749e05 z^5 + 2.059e05 z^4 - 2.852e05 z^3 - 3.421e05 z^2 + 1.205e05 z + 1.464e05}$$

5.4 Návrh číslicového filtru typu pásmová propust

Pásmová propust je vlastně spojení horní a dolní propusti. Máme tedy dvě frekvence řezu. Jednu menší, která funguje jako horní propust, a druhou větší fungující jako propust dolní.



Obr.5.5 Příklad Butterworth filtru typu PP

Při návrhu filtru typu PP volím vzorec pro Butterworthův filtr 2. řádu a to je

$$H(p) = \frac{1}{p^2 + \sqrt{2}p + 1}$$

Převodní vztah pro p je ovšem složitější, než u horní nebo dolní propusti.

$$p = \frac{p^2 + \omega_0^2}{p\omega_\Delta}$$

,kde $\omega_0^2 = \omega_{p_1}\omega_{p_2}$ a $\omega_\Delta = \omega_{p_2} - \omega_{p_1}$

Po převodu, úpravě a následní bilineární transformaci tedy získáváme

$$\frac{\left(\left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right) \omega_{\Delta}\right)^2}{\left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right)^4 + \left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right)^3 (\sqrt{2}\omega_{\Delta}) + \left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right)^2 (2\omega_0^2 + \omega_{\Delta}^2) + \left(\frac{2}{T_{vz}} \frac{z-1}{z+1}\right) (\sqrt{2}\omega_{\Delta}\omega_0^2) + \omega_0^4}$$

```
clearvars
clc

z=tf('z');

% Perioda vzorkovani
Tvz=0.005;

% Frekvence rezu
F1_cut = 4;
F2_cut = 6;

% Omegy s digitalni osou
omega_1 = F1_cut * 2 * pi;
omega_2 = F2_cut * 2 * pi;

% Omegy s analogovou osou
omega_1a = (2/Tvz)*tan(omega_1*Tvz/2);
omega_2a = (2/Tvz)*tan(omega_2*Tvz/2);

% Delta omegy
omega_delta = omega_2a - omega_1a;

% Power omega
omega_power = omega_1a * omega_2a;

% Bilinearni transformace
p = ((2/Tvz)*((z-1)/(z+1)));

% Filtr .... H(p) = 1 / (p^2 + sqrt(2)*p + 1)
% Pasmova Propust .... p => (p^2 + omega_power) /
p*omega_delta

Hz = (p*omega_delta)^2 /...
      (p^4 +...
      p^3*(sqrt(2)*omega_delta) +...
      p^2*(2*omega_power + omega_delta^2) + ...
      p * (sqrt(2)*omega_power + omega_delta) + ...
      omega_power^2)
```

Hz =

$$160000 z^5 + 160000 z^4 - 320000 z^3 - 320000 z^2 + 160000 z + 160000$$

$$1.749e05 z^5 + 2.059e05 z^4 - 2.852e05 z^3 - 3.421e05 z^2 + 1.205e05 z + 1.464e05$$

6 Tvorba programu na straně Arduina

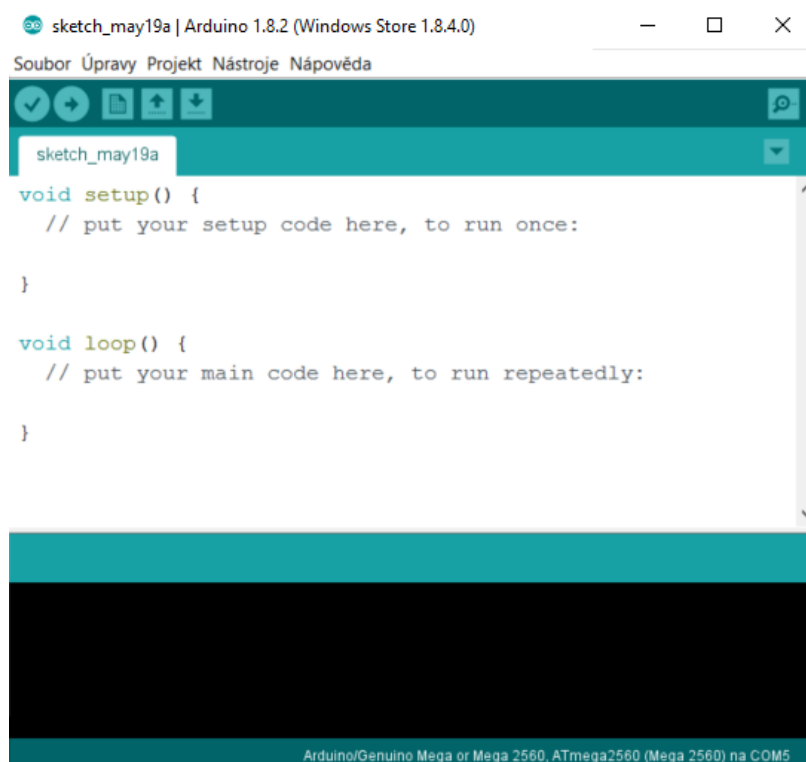
6.1 Arduino obecně

Arduino je otevřená platforma s grafickým vývojovým prostředím. Jedná se o desku s mikrokontrolerem a IDE. Desky Arduino obsahují 8bitové mikrokontrolery od firmy Atmel. Hlavní mikrokontroler je uživatelsky programovatelný, a to v jazyce podobném C/C++.

Díky otevřenosti projektu vzniká velké množství klonů Arduina a zároveň si ho lze poskládat z jednotlivých součástí na nepájivém poli.

6.2 Arduino IDE

Jak už jsem zmínil výše, Arduino má vlastní IDE, do kterého se píše programy v jazyce podobném C / C++. Tyto programy se skládají vždy ze dvou částí. První z nich je *void setup()*, což je část kódu, která se provede pouze na začátku slouží k přiřazení pinů, proměnných a podobně. Druhou částí programu je *void loop()*, do které se píše část kódu, jenž bude vykonáván pořád dokola.



Obr.6.1 Arduino IDE a dvě základní části programů

6.3 Program na straně Arduina týkající se zpracování signálu

V této části kapitoly ukážu a popíšu program pro zpracování signálu pouze pro jeden kanál z důvodu jednoduchosti popisu. Program s více kanály přijímaného signálu je přiložen na CD.

Můžete si všimnout, že, i když je požadovaná frekvence 200 Hz (perioda 5 milisekund), delay je 4,5 milisekundy. Je to z důvodu toho, že vykonání kódu také zabere nějaký čas. Tuto hodnotu jsem určil podle naměřené periody programu na osciloskopu.

Vytvoření proměnných

```
const int analogInPin = A0;

byte a = 0;

int u = 0;  // value read from the pot

float y = 0;  // value output to the PWM (analog out)


int ledState = LOW;
int ledPin = 53;


float y0 = 0;
float y1 = 0;
float y2 = 0;
float y3 = 0;


float u0 = 0;
float u1 = 0;
float u2 = 0;
float u3 = 0;
```

Inicializace programu s přenosovou rychlostí zápisu přes USB 9600 baudů

```
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
```


Inicializace opakujícího se programu

```
void loop() {
```

Touto částí je dosaženo pamatování 3 minulých hodnot

```
y3 = y2;
```

```
y2 = y1;
```

```
y1 = y0;
```

```
u3 = u2;
```

```
u2 = u1;
```

```
u1 = u0;
```

Čtení přijímaných hodnot z hardwaru

```
u = analogRead(analogInPin);
```

```
u0 = u * 0.0048828125;
```

Digitální filtr DP, jehož koeficienty jsou vypočítané Matlabem

```
y0=(float)((y1*143900 + y2*172300 - y3*146400 + u0*633.3 +  
u1*1900 + u2*1900 + u3*633.3)/174900);
```

Převedení čísla do 8bitové hodnoty, která je rychlejší pro přenos

```
a = (byte)(y0*50);
```

Odeslání hodnoty a počkání do požadovaného intervalu 5ms.

```
Serial.println(a);
```

```
delayMicroseconds(4500);
```

```
}
```

7 Program na straně Matlabu

Program na straně Matlabu je relativně jednoduchý, jelikož se jedná pouze o čtení a vykreslení hodnot. Do příkladu dám opět program pro zpracování signálu z pouze jednoho kanálu, abych zachoval integritu tištěné práce a bylo zřejmé, jaké hodnoty přijímá Matlab z Arduina. Vícekanálový program je opět přiložen na CD.

Zavře případně otevření čtení dat z COM

```
if (exist('s'))  
    fclose(s);  
end
```

Odstranění starých proměnných

```
close all  
clearvars  
clc
```

Otevření a zahájení čtení dat z COM5

```
s=serial('COM5','BaudRate',9600);  
fopen(s);
```

Zápis čtených hodnot po 3000 samplů a vykreslování grafu v reálném čase

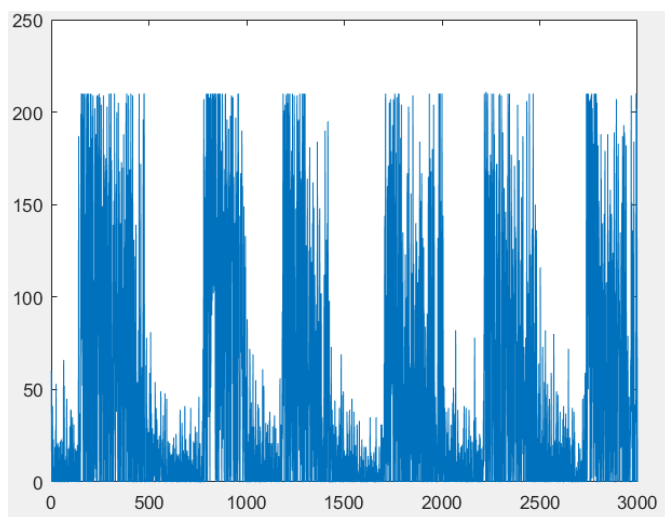
```
for i=1:3000  
    y(i)=fscanf(s,'%d');  
    if (mod(i,10)== 0)  
        pause(0.0001);  
        plot(y)  
        ylim([0 50])  
        xlim([0 3000])  
    end  
end
```

Uzavření COM

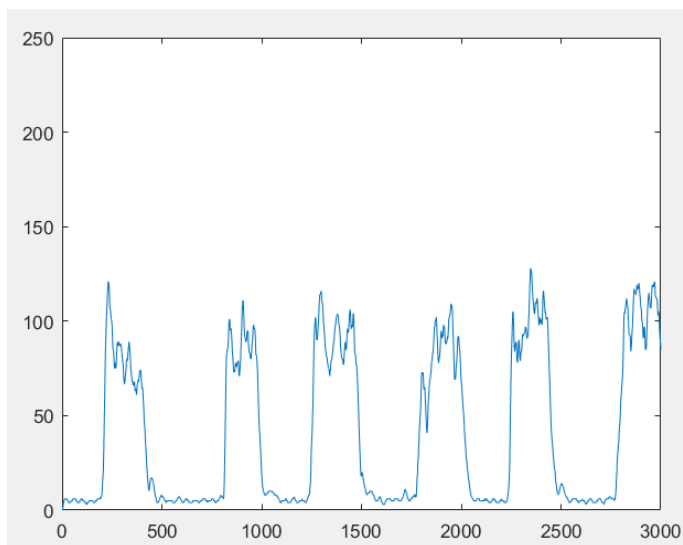
```
fclose(s);  
clc
```

7.1 Vykreslení výsledku zpracování signálu pro jeden kanál

Tyto výsledky testování jsou vykresleny v reálném čase v prostředí Matlab. Data jsou v tomto testování získávána přes diodu a nejsou ani podložena. V prvním grafu je zobrazen průběh signálu, který není upraven digitální filtrací v Arduino. Arduino zde slouží pouze jako A/D převodník. Jak je u prvního grafu vidět, maximální hodnoty přecházely do saturace, jinak řečeno dosáhli svého maxima. Je to způsobeno vstupním napětím, které je omezeno na 4,7 V. Zesilovačem tedy nemůžeme tuto hodnotu překročit. U druhého grafu se už signál v Arduino digitálně filtruje dolní propustí s frekvencí řezu 4 Hz a je vidět obálka signálu.



Obr.7.1 EMG signál bez digitálního filtru



Obr.7.2 EMG signál s digitálním filtrem typu DP

8 Závěr

Celý systém funguje bez problémů. Největším problémem bylo správně vytvořit hardwarovou část, nad kterou jsem strávil většinu času. Arduino MEGA2560 se osvědčilo jako dostatečný A/D převodník a zároveň mikrokontrolér tím, že převody a výpočty dále posílaných hodnot stíhalo bez problémů.

Celý systém byl vyzkoušen i na více lidech a hodnoty vykresloval správně.

Pro tuto práci je určitě spousta možností, jak pokračovat. Dalo by se vytvořit automatický regulátor zesílení a podložení pro různé signály, podle toho, kdo se nechává měřit jaký signál. Tento regulátor by také sloužil jako ochrana přepětí, aby se zařízení nepoškodilo. Dále by se mohlo toto řešení zapouzdřit a minimalizovat, aby to fungovalo jako autonomní biopack, který se naladí podle přijímaného signálu a mohl by mít vlastní displej nebo vlastní aplikaci v PC, protože prostředí Matlab je zbytečně komplexní nástroj pro funkci, kterou po něm zatím požadujeme.

Seznam použité literatury

- [1] SVATOŠ, Josef. Biologické signály I: geneze, zpracování a analýza. Vyd. 2. Praha: České vysoké učení technické, 1998. ISBN 8001018229.
- [2] VÍCH, Robert a Zdeněk SMÉKAL. Číslicové filtry. Praha: Academia, 2000. Česká matice technická, roč. 105 (2000), č. spisu 482. ISBN 802000761X.
- [3] BELZA, Jaroslav. Operační zesilovače pro obyčejné smrtelníky. Praha: BEN - technická literatura, 2004. ISBN 8073000601.